Basic Terms

The best way to understand the terms is to give an example for each term.

1. Graph (General)

   A **graph** $G = (V, E)$ consists of two sets V and E.

   - The elements of V are called **vertices** (or **nodes**).
   - The elements of E are called **edges**.
   - Each edge has a set of one or two vertices associated to it, which are called its **endpoints**. An edge is said to **join** its endpoints.

2. Edge-types

   - A **loop** is an edge that joins a single endpoint to itself.
   - A **multiple edge** is a collection of two or more edges having identical endpoints.

3. Finite and Infinite graph

   A **finite graph** is a graph with a finite number of vertices and edges. A graph which is not finite is called **infinite**.

4. Hypergraph

   A **hypergraph** is a generalization of a graph, where an edge can contain any number of vertices.

5. Multigraph and Pseudograph

   - A **multigraph** is a graph which is permitted to have multiple edges, that is, edges that have the same endpoints.
   - A **pseudograph** is a non-simple graph in which both graph loops and multiple edges are permitted.

6. Simple graph

   A **simple graph** is a graph that has no self-loops or multi-edges.

7. Subgraph and Supergraph

   A graph H is called a **subgraph** of graph $G = (V, E)$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. And G is a **supergraph** of H.

8. Edge-deletion

   $G = (V, E)$ is a graph, $e \in E$, then $G - e$ is defined by $G - e = (V, E \setminus \{e\})$.

9. Vertex deletion

   A graph $G = (V, E)$, $v \in V$, then $G - v$ is defined by $G - v = (V \setminus \{v\}, E')$, where $E' = \{e | e \in E$ and v is not the endpoint of $e\}$.

10. Induced subgraph

    Let $G = (V, E)$ be any graph, and let $S \subset V$ be any subset of vertices of G. Then the **induced subgraph** $< S >_G$ is the graph whose vertex set is S and whose edge set consists of all of the edges in E that have both endpoints in S..

11. Edge induced subgraph

    Each subset $E' \subseteq E$ defines a unique subgraph $H = (V', E')$ of graph $G = (V, E)$, where V' consists of only those vertices which are the endpoints of the edges in E'. The **edge-induced subgraph** H is denoted $< E' >_G$.

12. Neighborhood

    The **neighborhood** of a vertex v of a graph is the set of all vertices adjacent to v. It is denoted by $N(v)$.

13. Closed neighborhood

    The **closed neighborhood** is denoted $N[v] = N(v) \cup \{v\}$, where $N(v)$ is the neighborhood of a vertex v in a graph G.

14. Degree

    The **degree** (or **valency**) of a vertex v in a graph G, denoted $deg_G(v)$, is the number of proper edges incident on v plus twice the number of self-loops. (For simple graphs, of course, the degree is simply the number of neighbors.) The **maximum degree** of graph G is

$\triangle(G) = max\{deg(v) \mid v \in V(G)\}$ and the **minimum degree** of G is $\delta(G) = min\{deg(v)|v \in V(G)\}$.

15. Volume

    A graph $G = (V, E)$, the **volume** of G is $\sum_{v \in V(G)} deg(v)$.

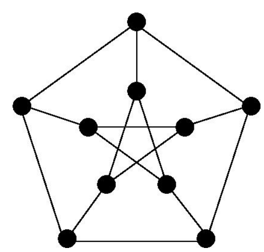
16. Regular graph

    A graph is **regular** if every vertex is of the same degree.

    It is **r-regular** if every vertex is of degree r. And r is called the **valency** of G.

17. Petersen Graph

    Petersen graph is an undirected 3-regular graph with 10 vertices and girth 5 as this figure.



18. Order and Size

    In a graph $G = (V, E)$, the **order** of G is the number of vertices in G, denoted $|G|$. And the **size** of G is the number of edges in G, denoted $||G||$.

19. Complete graph

    A **complete graph** is a graph in which every pair of vertices is joined by an edge.

20. Isolated vertex

    An **isolated vertex** in a graph is a vertex of degree 0.

21. Union and Intersection

    Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs.

    - $G = (V, E)$ is the **union** of G_1 and G_2, if $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$.
    - $G = (V, E)$ is the **intersection** of G_1 and G_2, if $V = V_1 \cap V_2$ and $E = E_1 \cap E_2$.

22. Decomposition

    If all the following are satisfy, then we said G_1,G_2,$\cdots$,G_t are the **decomposition** of G.

    - For all $i \in 1, 2, \cdots, t$, $V(G_i) \subseteq V(G)$.
    - $E(G_1) \cup E(G_2) \cup \cdots \cup E(G_t) = E(G)$.
    - For any $i \neq j$, $1 \leq i, j \leq t$, $E(G_i) \cap E(G_j) = \emptyset$.

23. Cartesian Product (Square product)

    The **cartesian product** (or square **product**) of two graphs G and H is denoted by $G\Box H$, where $V(G\Box H) = V(G) \times V(H)$ and two vertices (x, y) and (x', y') are adjacent if and only if either $x = x'$ and $\{y, y'\} = yy' \in E(H)$ or $y = y'$ and $\{x, x'\} = xx' \in E(G)$.

24. Join

    The **join** (or **suspension**) of two graphs G and H is denoted by $G \vee H$, where

    $V(G \vee H) = V(G) \cup V(H)$ and

    $E(G \vee H) = E(G) \cup E(H) \cup \{uv|u \in V(G) \text{ and } v \in V(E)\}$.

25. Composition

    The **composition** $G[H]$ of a graph G with a graph H is the graph with vertex set $V(G) \times V(H)$ such that (u_1, v_1) is adjacent to (u_2, v_2) whenever either u_1 is adjacent to u_2, or v_1 is adjacent to v_2 with $u_1 = u_2$.

26. Complement

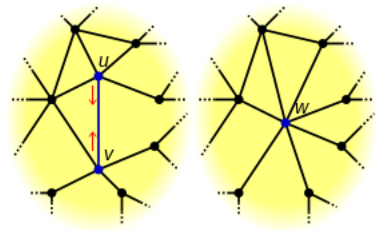
    The **complement**(or **edge-complement**) $\overline{G} = (V, \overline{E})$ of a simple graph $G = (V, E)$ has the same vertex set V as G and edges defined : $\{e\}$ is in $\overline{E}$ if and only if $\{e\}$ is not in E.

27. Line graph

    The **line graph** $L(G)$ of a graph G has the edge set of G as its vertex set, i.e. $V(L(G)) \equiv E(G)$; two vertices of $L(G)$ are adjacent if the edges in G to which they correspond have a common vertex, i.e. $E(L(G)) = \{ef|e, f \in E(G) \text{ and } |e \cap f| = 1\}$.

28. Edge Contraction

    Let $G = (V, E)$ be a graph containing an edge $e = uv$ with $u \neq v$. Let f be a function which maps every vertex in $V \setminus \{u, v\}$ to itself, and otherwise, maps it to a new vertex w. The contraction of e results in a new graph $G' = (V', E')$, where $V' = (V \setminus \{u, v\}) \cup \{w\}$, $E' = E \setminus \{e\}$.



29. Contraction graph

    H is a **contraction graph** of G if H is obtained by contracting some edges in G.

30. Minor

    A **minor** of a graph G is a graph which can be obtained from G by deleting some vertices and edges, and then contracting some further edges.

31. Isomorphism

    An **isomorphism between two simple graphs** G and H is a vertex bijection $\phi : V(G) \to V(H)$ such that for $u, v \in V(G)$, the vertex u is adjacent to the vertex v in graph G if and only if $\phi(u)$ is adjacent to $\phi(v)$ in graph H. Implicitly, there is also an edge bijection $E(G) \to E(H)$ such that $uv \mapsto \phi(u)\phi(v)$. We say that G and H are **isomorphic graphs** and we write $G \cong H$.

32. Degree sequence

    The **degree sequence** of a graph G is the sequence of its degrees of vertices, arranged from the largest value to the smallest.

33. Automorphism

    Given a graph X, a permutation α of $V(X)$ is an **automorphism** of X if

    $$\{u, v\} \in E(X) \Leftrightarrow \{\alpha(u), \alpha(v)\} \in E(X), \text{ for all } u, v \in V(X).$$

34. Self-complementary

    A **self-complementary** graph is a graph which is isomorphic to its complement.

35. Graphical sequence

We call a sequence of non-negative integers $d_1, \dots, d_n$ **graphical** if there exists a graph G of order n the vertices of which have, in some order, degrees $d_1, \dots, d_n$.

36. Distance

The **distance** between two vertices in a graph is the length of the shortest walk between them. If there are no path between two vertices, then the distance of these two vertices is infinite.

37. Eccentricity

The **eccentricity** of a vertex v in a connected graph is its distance to a vertex farthest from v. Denote $e_G(v) = \max\{d_G(u, v)|u \in V(G)\}$.

38. Diameter and Radius

The **diameter** of a connected graph G is its maximum eccentricity. Denote

$$D(G) = \max\{e_G(v)|v \in V(G)\}.$$

The **radius** of a connected graph G is its minimum eccentricity. Denote

$$r(G) = \min\{e_G(v)|v \in V(G)\}.$$

39. Center

The **center** of a graph is the subgraph induced on its set of the vertices whose eccentricity equals the radius of the graph.

40. Connected graph

A graph is **connected** if between every pair of vertices there is a walk. If a graph is not connected, then it is **disconnected**.

41. Acyclic, Forest, and Tree

- A graph is **acyclic** if it has no cycles.
- A **forest** is an acyclic graph.
- A **tree** is an acyclic connected graph.

42. Spanning subgraph and Spanning tree

- A subgraph H of a graph G is a **spanning subgraph** if $V(H) = V(G)$.
- A **spanning tree** of a graph G is a spanning subgraph of G that is a tree.

43. Component and Trivial component

- A **component** of a graph G is a connected subgraph H such that no subgraph of G that properly contains H is connected. In other words, a component is a *maximal* connected subgraph.
- A component (or graph) is called **trivial** if it consists of one vertex.

44. Bridge graph and Bridgeless graph

- A **bridge** of a connected graph is a graph edge whose removal disconnects the graph.
- A **bridgeless graph** is a graph that contains no graph bridges.

45. Digraph

A **digraph** (or **directed graph**) is a graph each of whose edges is directed. An arc (x, y) is considered to be directed from x to y; y is called the **head** and x is called the **tail** of the arc; y is said to be a direct **successor** of x and x is said to be a direct **predecessor** of y.

46. Outdegree and Indegree

D is a digraph.

- The **outdegree** of a vertex v in D is the number of arcs directed from v.
- The **indegree** of vertex v is the number of arcs directed to v.

47. Weakly connected, Strongly connected, Strong component

- A digraph is **weakly connected** if its underlying graph is connected.
- A digraph is **strongly connected** if every two vertices are mutually reachable.

- A **strong component** of a digraph G is a maximal strongly connected subdigraph of G.

48. Orientation, Oriented graph, and Tournament

    An **orientation** of an undirected graph assigns a unique direction to each edge. An **oriented graph** is a digraph obtained by choosing an orientation for each edge of an undirected simple graph. A **tournament** is an oriented complete graph.

49. King

    A **king** in a tournament T is a vertex x with maximum out-degree.

50. Weighted graph

    A **weighted graph** is a graph in which each edge is given a numerical weight. A weighted graph is therefore a special type of labeled graph in which the edge labels are numbers (which are usually taken to be positive).

51. Adjacency matrix and Incidence matrix

    - An **adjacency matrix** for a simple graph G whose vertices are explicitly ordered $v_1, v_2, \cdots, v_n$ is the $n \times n$ matrix A_G such that

    $$A_G(i,j) = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are adjacent; and} \\ 0, & \text{otherwise.} \end{cases}$$

    - An **incidence matrix** for a simple graph G whose vertices are explicitly ordered $v_1, v_2, \cdots, v_n$ and edges are explicitly ordered $e_1, e_2, \cdots, e_m$ is the $n \times m$ matrix B_G such that

    $$B_G(i,j) = \begin{cases} 1, & \text{if } v_i \text{ and } e_j \text{ are incident; and} \\ 0, & \text{otherwise.} \end{cases}$$

52. Cycle matrix

    If $|E(G)| = m$, and $C_1, C_2, \ldots, C_q$ are all the circuits in graph G. Then the **cycle matrix** of G is a matrix $C(G) = (a_{ij})$ with $q \times m$, and

    $$a_{ij} = \begin{cases} 1, & \text{if } e_j \text{ is an edge of } C_i, \text{ and} \\ 0, & \text{if } e_j \text{ otherwise.} \end{cases}$$